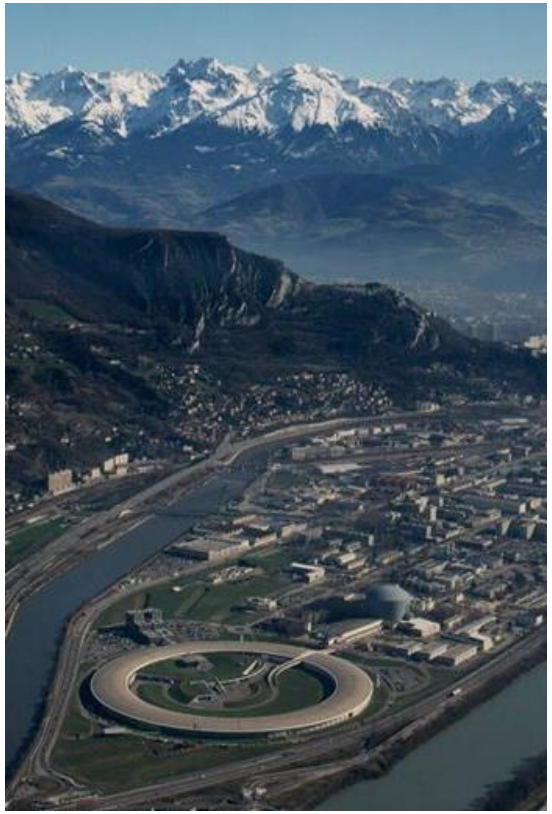

ESRF
The European Synchrotron

BLISS AND 2D DETECTOR DAQ



Outline

- **Introduction to BLISS**
- **LIMA Control & DAQ**
- **Scanning and Synchronization**

Jens Meyer on behalf of the ESRF Beamline Control Unit

GENERAL CONCEPTS

Command line driven data acquisition sequencer written in Python

Main concepts

- **Hardware abstraction layer** for all instrumentation used during a data acquisition sequence

  Motors, counters, monochromators, spectrometers, diffractometers, 2D detectors, etc.

- **A generic scan engine** for step and continuous scans

  The use of trajectories and HKL space is possible with all scans

- **Decoupling of data acquisition** from data saving and analysis

  All data buffered in memory. Allows higher acquisition speed without blocking

- **Coherent HDF5 storage** of all acquired data at high speed and for large data volumes

  All data of a proposal, its samples and the produced datasets is saved as a coherent HDF5 data tree

- **Live data display** of all acquired data

  Immediate visibility of acquisition results for the user

- **Easy configuration** of hardware and experimental environment

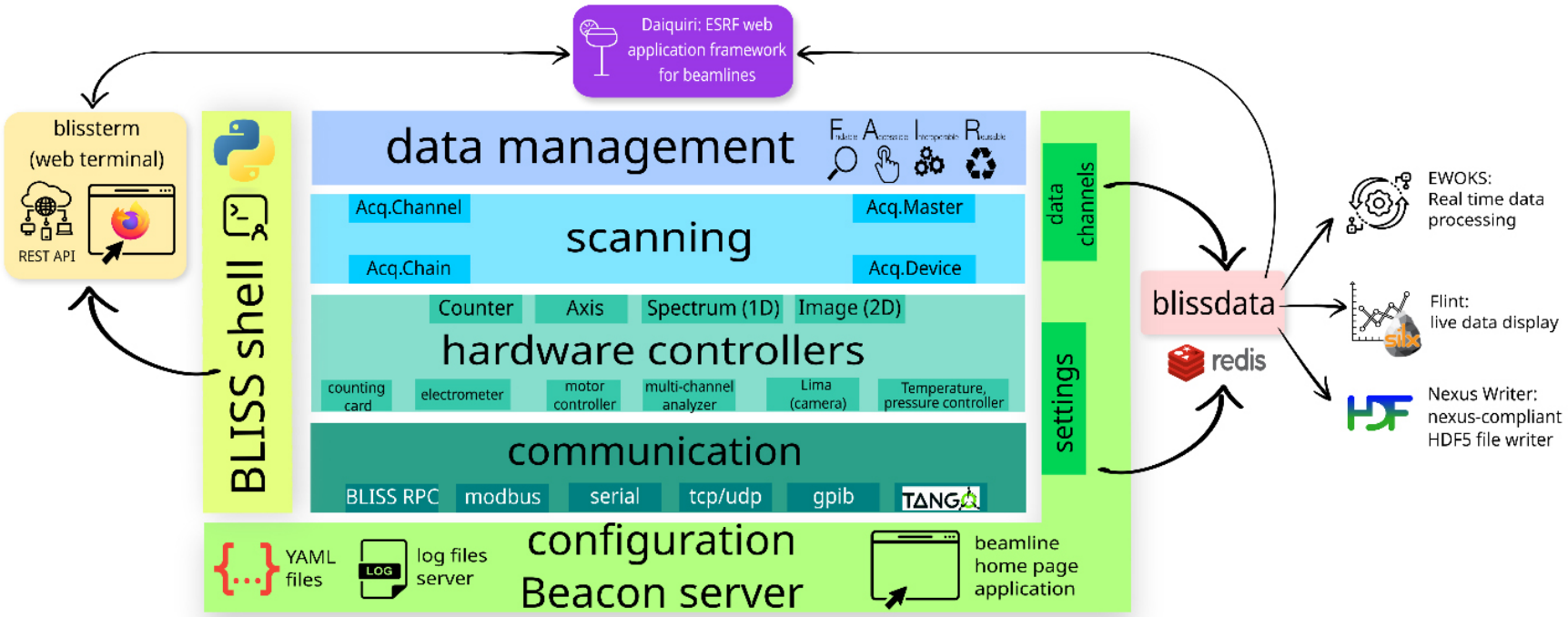
  Switch between predefined acquisitions set-ups on the fly

- **PyTango** to interface any device from the Tango world

  https://www.tango-controls.org

https://bliss.gitlab-pages.esrf.fr/bliss/master

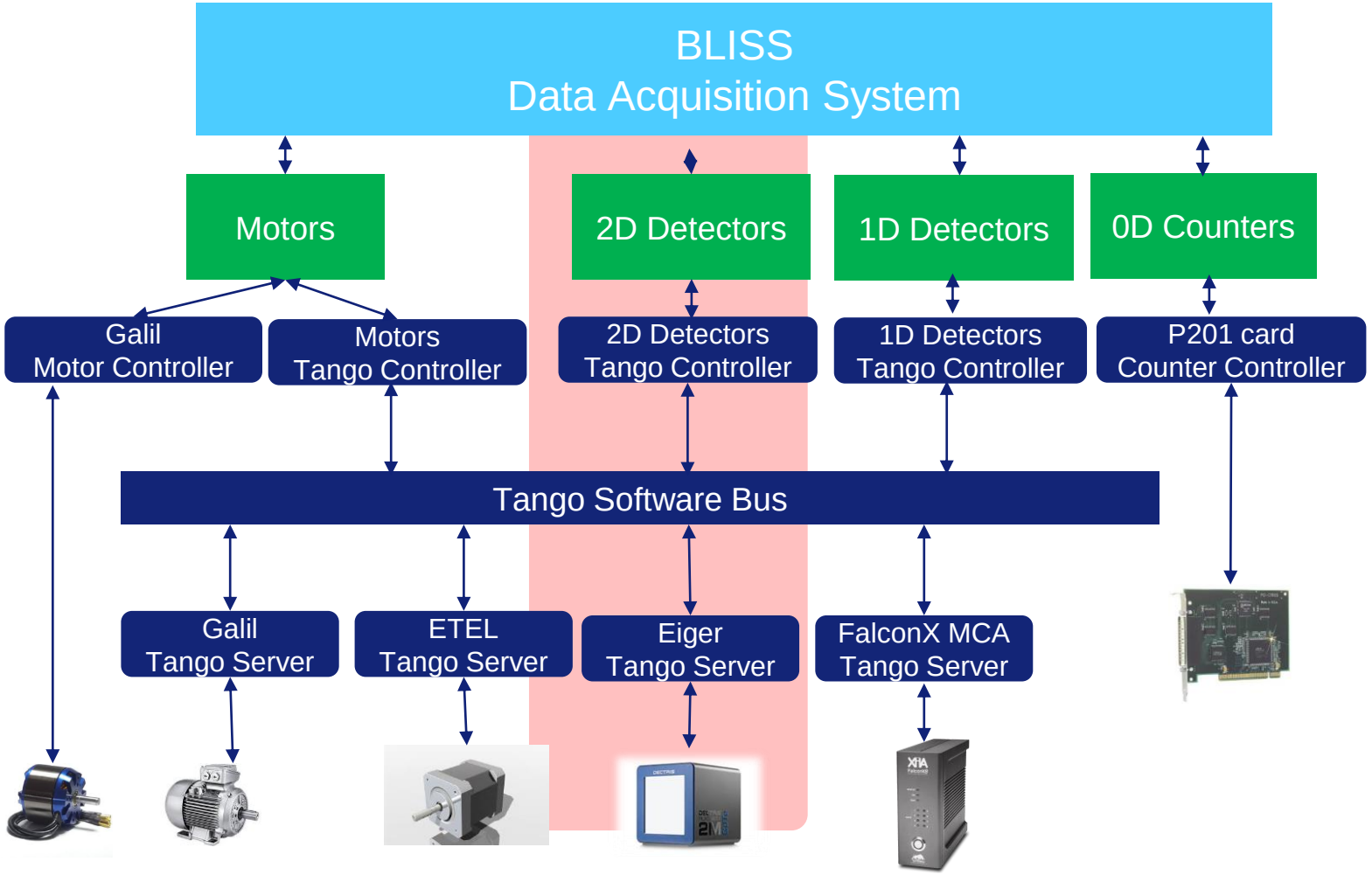
Schematic view of the BLISS software layers



HARDWARE ABSTRACTION

Direct hardware access or underlying control system?

All 2D detectors integrated with LIMA and wrapped as Tango Server



BLISS provides easy configuration of LIMA subsystems

- Image transformation
- Acquisition mode / accumulation
- Pixel mask
- Background subtraction
- Flat-field correction

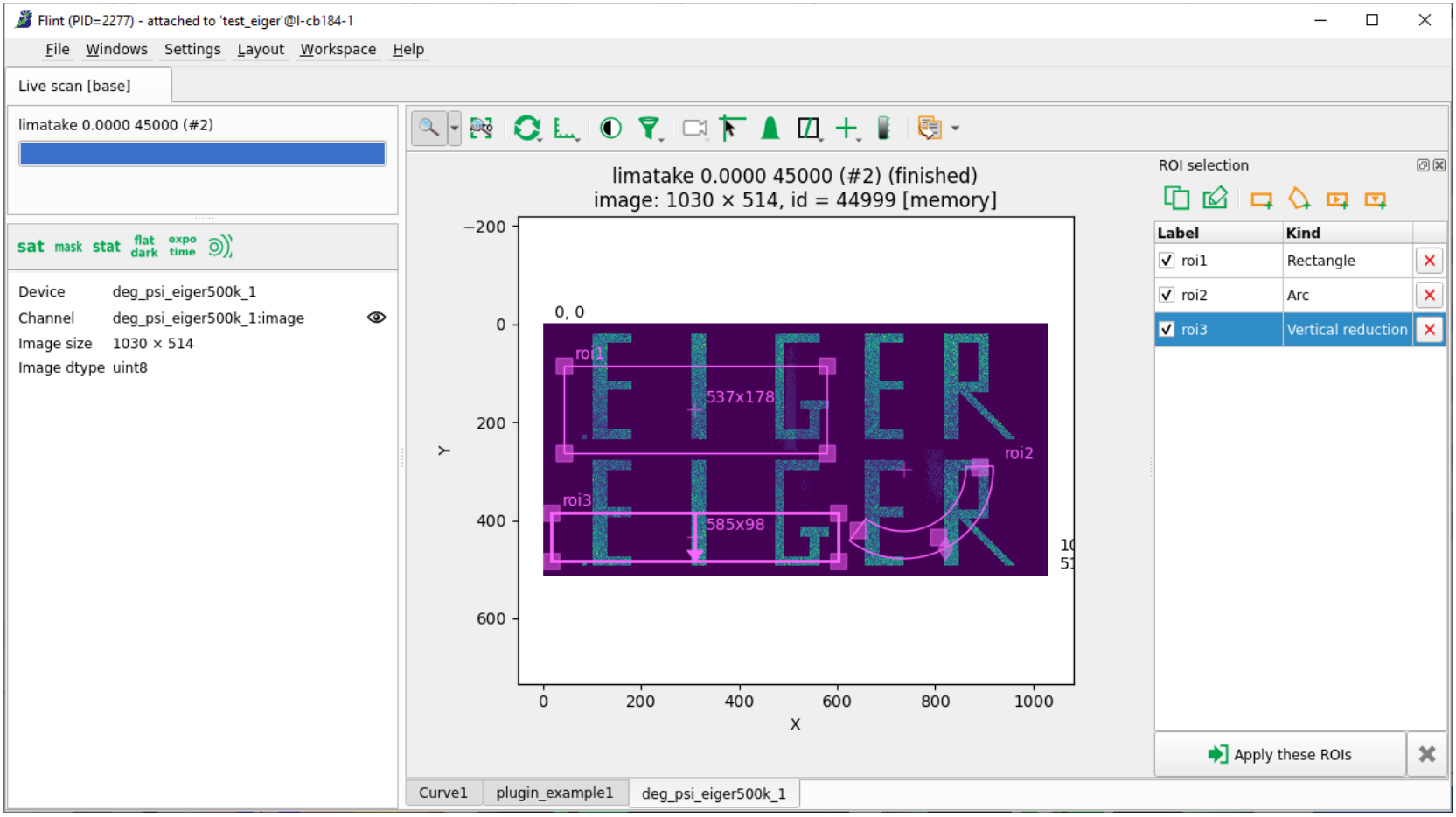
```
TEST_EIGER [6]: deg_psi_eiger500k_1
       Out [6]: deg_psi_eiger500k_1 - PSI/Eiger-500k (SlsDetector) - Lima SlsDetectorEiger

                Image:
                size (w,h): 1030, 514  full size (1030, 514)
                depth:      1
                bpp:        Bpp8
                binning:    [1, 1]
                flip:       [False, False]
                rotation:   0
                roi:        None

                Acquisition:
                status: |'Ready'|
                status_fault_error: |'No error'|
                tag: 4294967295
                mode: 'SINGLE'
                nb_frames: 45000
                expo_time: 1e-05
                trigger_mode: 'INTERNAL_TRIGGER'

                Mask
                ----
                use mask: False
                mask image path: ** UNSET **
```

- **Saving parameters (including expert settings):**

```
    Saving
--------------
File Format:   HDF5BS
 └-> Suffix:   .h5
Current Mode:  ONE_FILE_PER_N_FRAMES
Available Modes:
 - ONE_FILE_PER_FRAME
 - ONE_FILE_PER_SCAN
 - ONE_FILE_PER_N_FRAMES
 - SPECIFY_MAX_FILE_SIZE

for ONE_FILE_PER_N_FRAMES mode
------------------------------
frames_per_file: 10000

for SPECIFY_MAX_FILE_SIZE mode
------------------------------
max file size (MB):  500
 └-> frames per file: 990

Expert Settings
---------------
config max_writing_tasks:  1
current max_writing_tasks: 1
lima managed_mode:         SOFTWARE
```

Saving location is defined by the high-level ESRF data policy

- **RoI counter & ROI-2-Spectrum selection in Flint**



The European Synchrotron | ESRF

LIMA CONTROL

- **Online data reduction:**

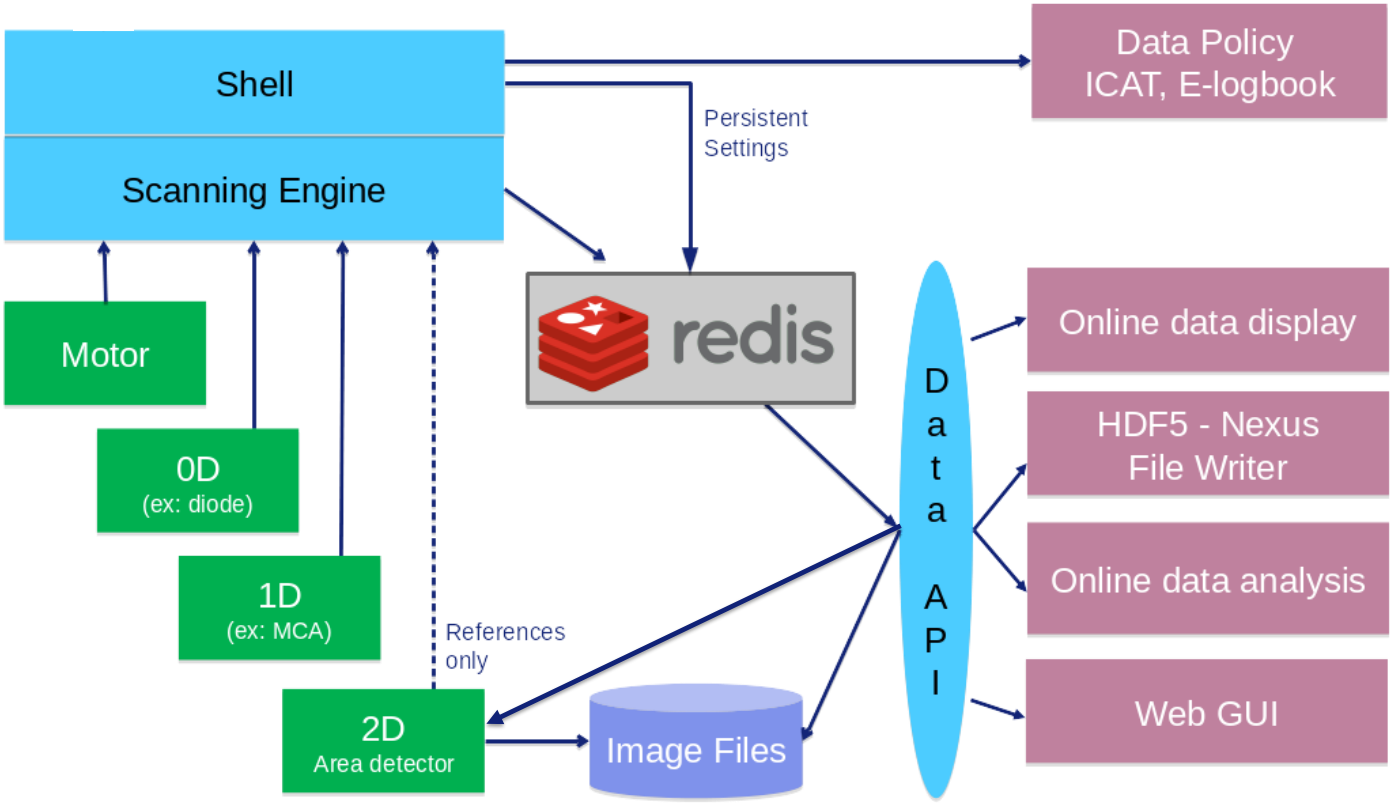
```
ROI Counters: default
   Name                    Parameters                      State
--------- ------------------------------------------- ----------
   roi1                <42,84> <537 x 178>              Enabled
   roi2    <735.5, 295.1> <127.2, 183.2> <-1.7, 126.9>  Enabled

Roi Profile Counters: default
   Name          <x, y> <w, h> <mode>         State
--------- ------------------------------- ----------
   roi3    <17,384> <585 x 98> <horizontal>  Enabled

ROI Collection: default
*** no ROI defined ***

BPM Counters:
acq_time, intensity, x, y, fwhm_x, fwhm_y
```

- **Detector specific parameters:**

```
TEST_EIGER [7]: deg_psi_eiger500k_1.camera
       Out [7]: apply_corrections: False
                signed_image_mode: False
                threshold_energy: -1
                all_trim_bits: array([-1, -1], dtype=int32)
                high_voltage: 150
                clock_div: 'FULL_SPEED'
                fixed_clock_div: False
                parallel_mode: 'PARALLEL'
                skip_frame_freq: 0
                tx_frame_delay: 0
```

- **And more!**

Decoupling of data acquisition from data saving and analysis



- LIMA 0D & 1D data is directly injected into Redis
- Only References for 2D data
  - Reading “warm frames” from LIMA server memory
  - Reading “cold frames” from file when LIMA memory is overwritten

BLISS DATA

A high-level interface to the data in the BLISS ecosystem

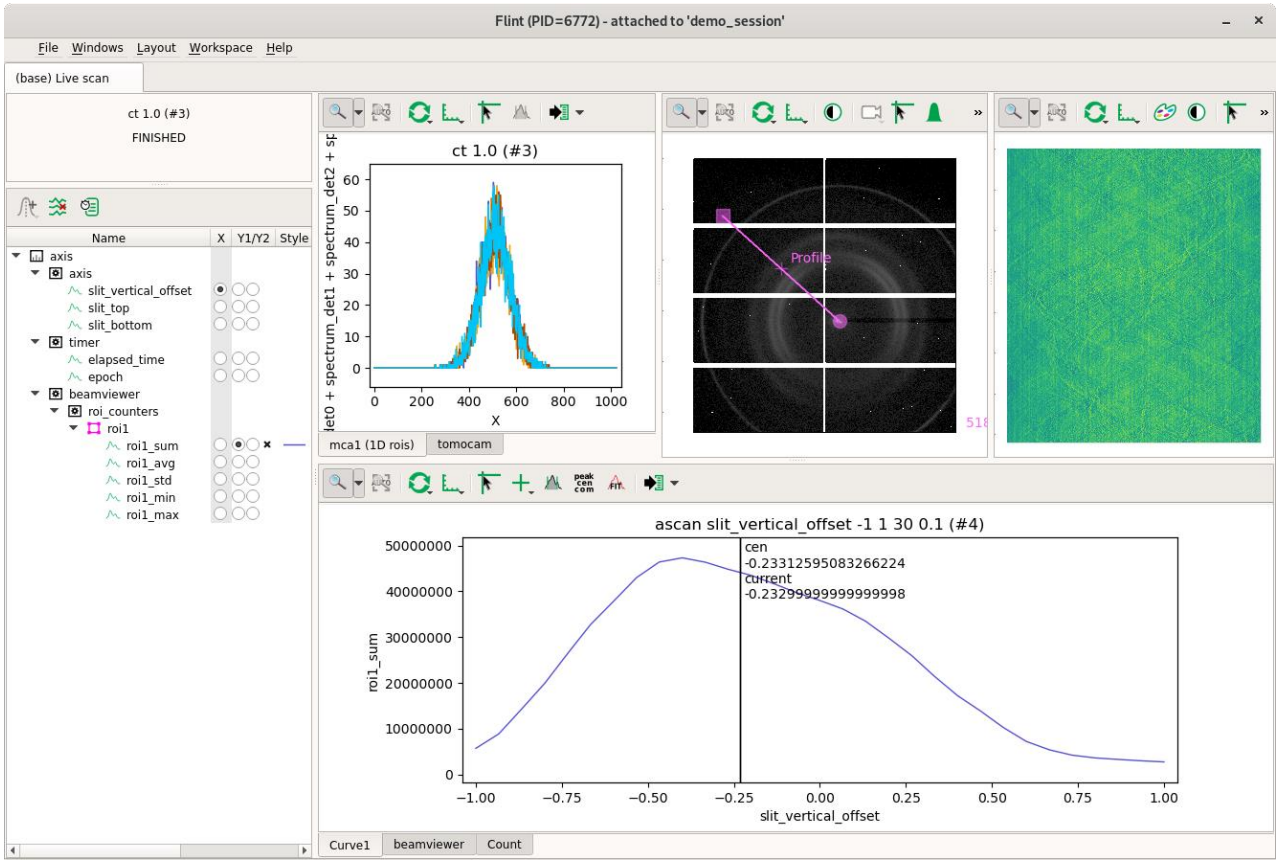
- Data is injected from the source through streams
- Client register to streams and are updated as soon as new data arrives

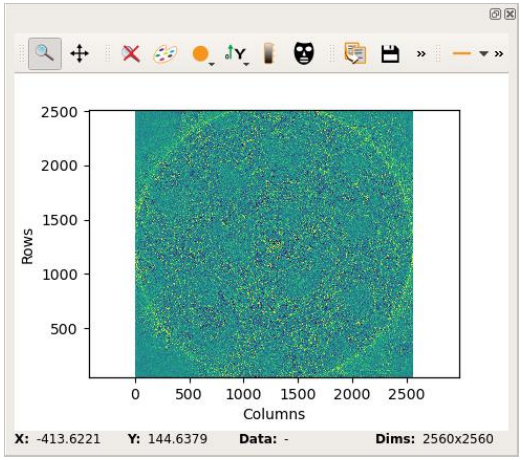
```
scan = load a scan from blissdata
lima_stream = scan.streams["lima:image"] # lima_stream

# access by index directly
try:
    lima_stream[45]
except IndexErrorNotYetThere:
    return # image 45 is not yet acquired

# access available data sequentially
cursor = lima_stream.cursor()
while True:
    try:
        view = cursor.read()
    except EndOfStream:
        return # nothing more to read
    images = view.get_data() # download all the new images
```

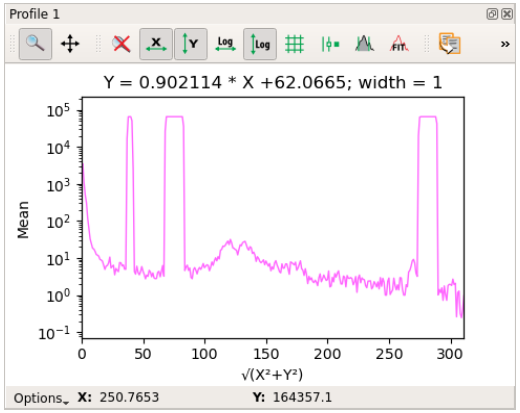
Flint - Live Data Display

Live scan data

Calculated data can be pushed to Flint

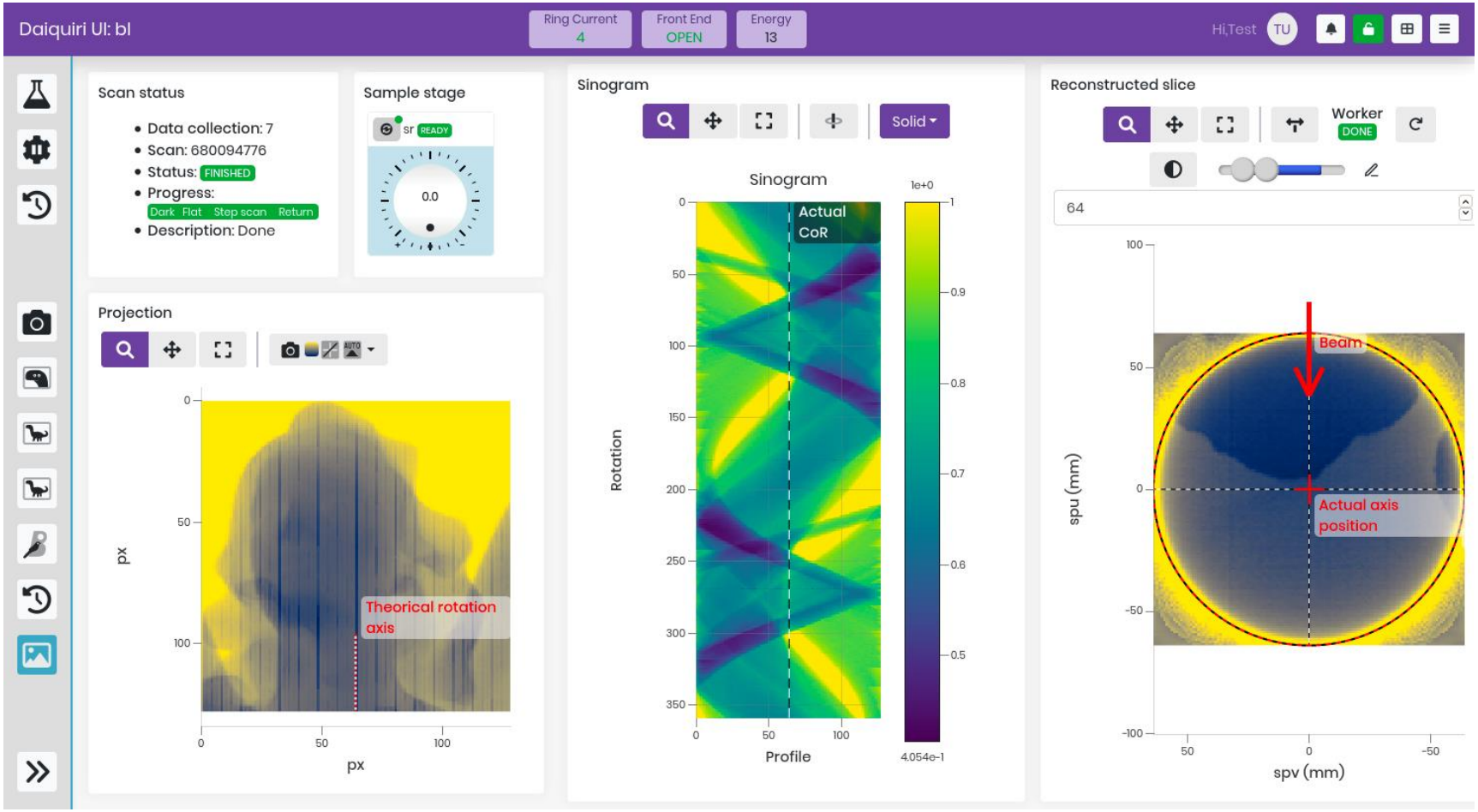






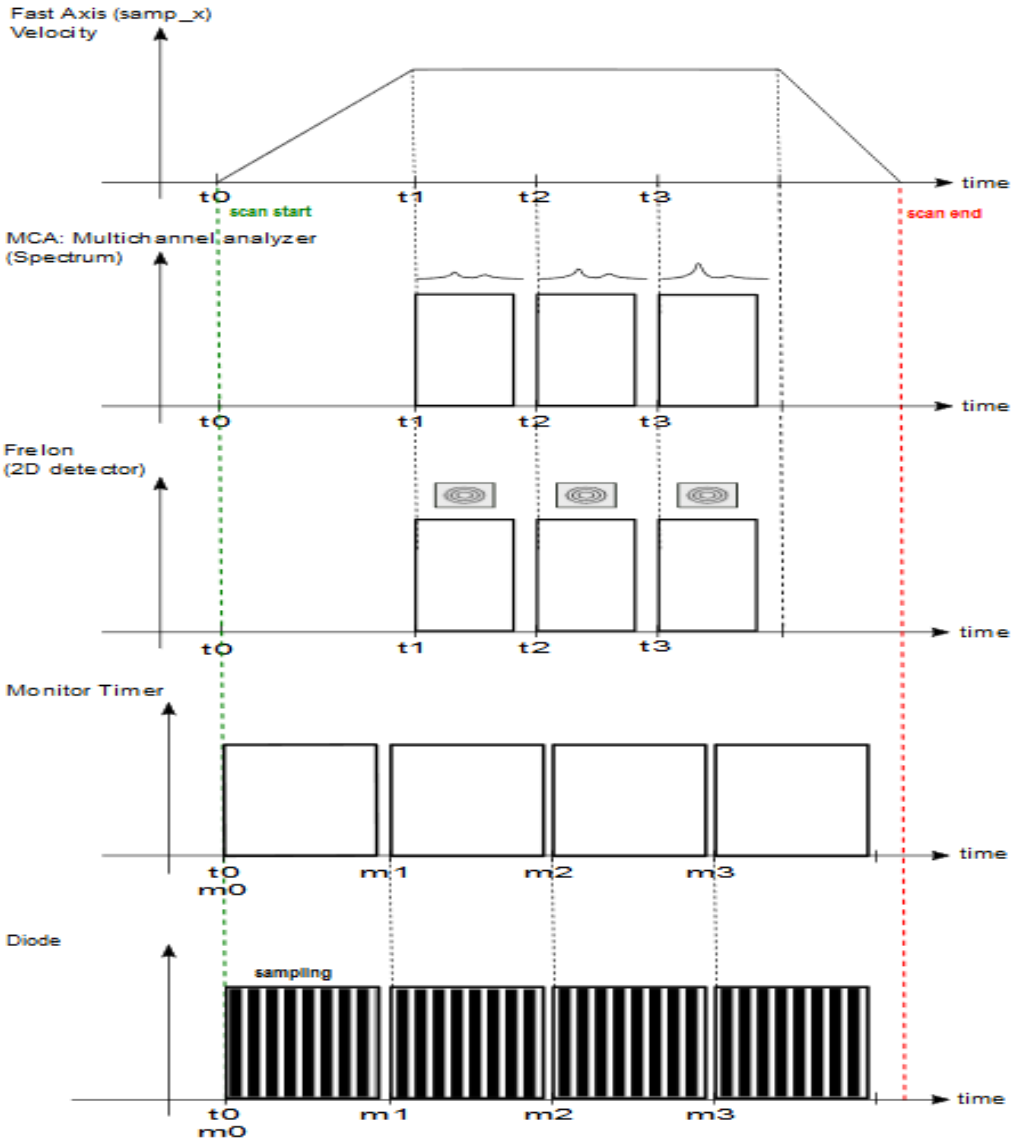
Daiquiri for Tomography

Tomography sinograms are built from Roi-2-Spectrum 1D data



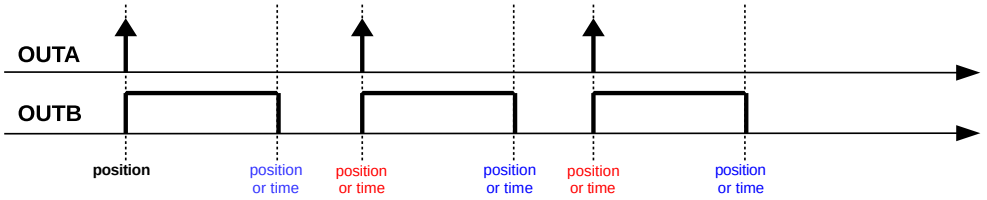
BUILDING A CONTINUOUS SCAN

- Motor:samp_x
  - MCA
    - spectrum
  - Lima
    - image
- Timer
  - Keithley
    - scalar

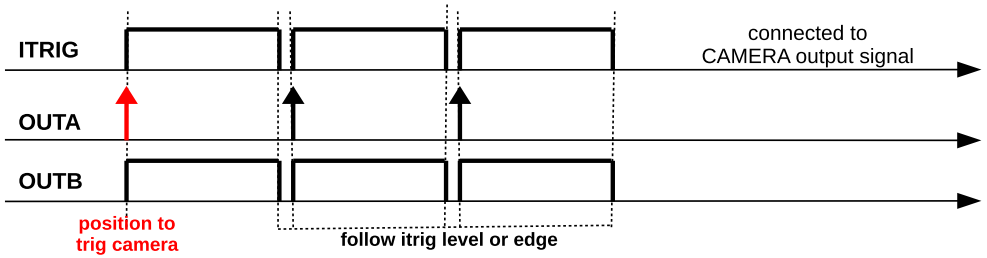


FSCAN TRIGGER MODES

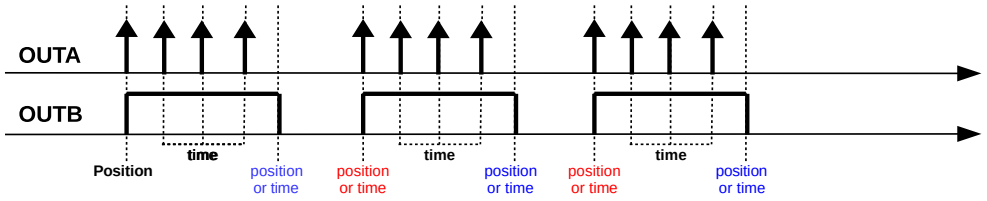
***TIME* or *POSITION* mode**



***CAMERA* mode**



LIMA accumulation
supported in all modes



CONCLUSIONS

- The LIMA interface is powerful but complex
- A high-level client library is missing for LIMA1
- BLISS fills this gap to make its features easy to use
- With LIMA2 a more complete high-level client library is under development

THANK YOU

Acknowledgements to

- The members of the software group for the development of all the different software tools

- Any Questions?